Overriding Subsuming Rules

Philippe Besnard[1], Éric Grégoire[2], and Sébastien Ramon[2]

[1] IRIT, F-31062 Toulouse
CNRS UMR 5505, F-31062
118 route de Narbonne, F-31062 Toulouse France
besnard@irit.fr
[2] Université Lille - Nord de France, Artois, F-62307 Lens
CRIL, F-62307 Lens
CNRS UMR 8188, F-62307
rue Jean Souvraz SP18, F-62307 Lens France
{gregoire,ramon}@cril.univ-artois.fr

Abstract. This paper is concerned with intelligent agents that are able to perform nonmonotonic reasoning, not only *with*, but also *about* general rules with exceptions. More precisely, the focus is on enriching a knowledge base Γ with a general rule that is subsumed by another rule already there. Such a problem is important because evolving knowledge needs not follow logic as it is well-known from e.g. the belief revision paradigm. However, belief revision is mainly concerned with the case that the extra information logically conflicts with Γ. Otherwise, the extra knowledge is simply doomed to extend Γ with no change altogether. The problem here is different and may require a change in Γ even though no inconsistency arises. The idea is that when a rule is to be added, it might need to override any rule that subsumes it: preemption must take place. A formalism dedicated to reasoning with and about rules with exceptions is introduced. An approach to dealing with preemption over such rules is then developed.

Keywords: Dynamics of knowledge, Logic, Default reasoning.

1 Introduction

Assume a knowledge base Γ contains the rule *If the switch is on then the light is on*. When *If the switch is on and the lamp bulb is ok then the light is on* needs to be introduced inside Γ, it seems natural to require this new rule to preempt the older rule: it is no longer enough to know that the switch is on to be able to conclude that the light is on, it must additionally be the case that Γ yields the information that the lamp bulb is ok. First, let us observe that a monotonic logic cannot capture such dynamics of reasoning by simply adding the new rule. According to monotonicity, any conclusion drawn from a given set of premises can still be inferred whatever additional premises happen to supplement this set. In such a logic, the statement *the light is on* (concluded from the former rule and the statement *the switch is on*) is still concluded even though the second rule is added, and, worse yet, regardless of any information stating that the lamp bulb is broken. Also, the usual approaches to belief revision [AGM85] fail to address this issue because they make the new information to be set-theoretically unioned with Γ in case no inconsistency arises. Let us stress that moving to a nonmonotonic formalism where

exception to rules depends on consistency checks like adding *If the switch is on and if it can be consistently assumed that the lamp bulb is ok, then the light is on* does not change the problem.

Technically, the problem can be described as follows. Given a set Γ of formulas and a rule R, what changes should Γ undergo so as to infer R but not to infer any R' subsuming R? In symbols, where $\Gamma^\star$ stands for Γ after these changes have taken place,

$$\Gamma^\star \mid\!\sim R \qquad \text{and} \qquad \Gamma^\star \not\mid\!\sim R'$$

Clearly, the problem first requires several matters to be settled. First, the syntax for rules (in which $R, R', \ldots$ are expressed) is to be defined. Second, an inference relation (denoted $\mid\!\sim$) allowing rules to be handled needs to be settled. Third, a concept of implicant for rules expressing what does R' subsuming R mean needs to be proposed, before an approach to solve the above preemption issue can be defined.

Accordingly, the paper is organized as follows. In the next Section, a general formalism for representing rules with exceptions is introduced with the aim of encompassing various logic-based approaches allowing such rules, including default reasoning. Section 3 introduces useful inference tools to reason about such rules, while Section 4 connects the tools with default logic. In Section 5, a useful X-derivation concept is proposed, allowing both plain formulas and rules to be inferred under the possible assumption of additional formulas or other rules with exceptions. Section 6 investigates a concept of implicant for rules with exceptions. The approach to the preemption issue is then developed in Section 7, based on the X-derivation and the latter implicant concepts. Finally, some avenues for future research are provided in the conclusion.

Throughout the paper, the following notations are used: $\neg$, $\vee$, $\wedge$ and $\supset$ denote the classical negation, disjunction, conjunction and material implication connectives, respectively. When Ω is a set of formulas, $Cn(\Omega)$ denotes the deductive closure of Ω under a given logic, of which $\|\!-$ denotes the consequence relationship, $\bot$ denotes absurdity, and $\top$ denotes any tautology.

2 Rules with exceptions

2.1 Defaults

In the Artificial Intelligence research community, some of the most popular tools to handle forms of defeasible reasoning remain rules with exceptions, e.g., in the form of defaults [Rei80]. They permit an inference system to jump to default conclusions and to withdraw them when new information shows that these conclusions now lead to inconsistency. Usually, such a rule is based on logical formulas, that is, expressions of a formal language upon which an inference system (no matter how poor or rich) models some kind of reasoning.

For instance, in default logic [Rei80], a *default* is of the form:

$$\frac{\alpha : \beta}{\gamma}$$

where α, β, γ are formulas of classical logic.

Intuitively, such a default is intended to allow the reasoning "*Provided that α is inferred and provided that β is consistent w.r.t. what is inferred, infer γ*".

Importantly, the inference notion alluded to is based on a logic (Reiter made it to be classical logic but it is possible to have another logic instead: see paraconsistent default logic [PB91] for example).

2.2 PEC rules

Let us first concentrate on rules with exceptions under consistency tests. Our leading example is then expressed as *If the switch is on and if it can be consistently assumed that the lamp bulb is ok, then the light is on* and consists of three parts: its premises, its exceptions, and its conclusions. E.g., it could be represented by the default

$$\frac{switch\_on : lamp\_bulb\_ok}{light\_on}$$

We aim at representing such rules in a uniform way within a unified setting that is, among other things, meant to be general enough as to encompass default logic while allowing us to instantiate it to other logical formalisms. It should also allow the representation of both monotonic knowledge and rules involving consistency checks.

Given a logical language, a PEC rule (for Premises-Exceptions-Conclusions) is a triple consisting of three sets of formulas. First, the premises, which are the necessary conditions for *this* rule to apply. Then, the exceptions, which are based on consistency tests. Finally, the conclusions, which list the claims that can be made whenever the rule applies.

Definition 1. *A PEC rule is a triple $\mathcal{R} = (\mathcal{P}, \mathcal{E}, \mathcal{C})$ where $\mathcal{P} = \{\rho_1, \ldots, \rho_k\}$ and $\mathcal{C} = \{\varsigma_1, \ldots, \varsigma_n\}$ are consistent sets of formulas and $\mathcal{E} = \{\epsilon_1, \ldots, \epsilon_m\}$ is a set of non-tautological formulas.*

Importantly, we impose no constraint on the language: there may, or may not, be connectives such as negation, conjunction, disjunction and the like. There may even be no connective at all. However, an underlying inference relation $\|\!\!-$ must be available. Of course, this means that the logical formalism used must have a form of tautology (please note the subtlety here: this does not mean that the logical formalism used must have tautologies!).

A PEC rule can be interpreted in different ways, depending on how its set of premises and its set of conclusions are captured logically (presumably, conjunctively or disjuntively). In the sequel, we consider only unary PEC rules, that is, PEC rules whose sets of premises and sets of conclusions are singletons. Abusing the notation in order to improve readability, we often omit curly brackets for these singletons.

Definition 2. *A PEC rule $\mathcal{R} = (\mathcal{P}, \mathcal{E}, \mathcal{C})$ is unary iff $\mathcal{P}$ and $\mathcal{C}$ are singleton sets.*

Example 1. The PEC rule $(switch\_on, \{\neg lamp\_bulb\_ok\}, light\_on)$ is an encoding of the rule with exception *If the switch is on and, consistently assuming that the lamp bulb is ok, then the light is on*.

Example 2. The PEC rule $(\{switch\_on, lamp\_bulb\_ok\}, \emptyset, light\_on)$ is an encoding of a similar rule where the impossibility to derive e.g. $lamp\_bulb\_ok$ can block the inference of $light\_on$. In this respect, $\neg lamp\_bulb\_ok$ would be an exception to the rule, which is however not to be included in the set of exceptions of the PEC rule, since it is not a consistency-based exception.

Example 3. The PEC rule $(\top, \emptyset, light\_on)$ is an encoding of the fact *the light is on*.

Example 4. The PEC rules $(switch\_on, \emptyset, light\_on)$, $(switch\_on \supset light\_on, \emptyset, \emptyset)$ and $(\emptyset, \emptyset, switch\_on \supset light\_on)$ are various encodings of the exception-free rule *If the switch is on then the light is on*.

As can be seen in the examples, exceptions to a rule that are supposed to be derived in the monotonic fragment of the logic (vs. consistency checks) are not included in the set of exceptions in the PEC rule, which is devoted to exceptions based on consistency checks. As regards exception-free statements, we represent them as PEC rules whose sets of exceptions are empty; regarding the premises, various choices are possible (e.g., the set of premises being a tautological singleton). In this respect, it follows that $\mid\!\!\vdash$ is assumed to admit $\top$ to represent effectively some formula. The various possible encodings of knowledge between premises and conclusions is similar to the well-known difference in default logic between defaults with prerequisites and the corresponding prerequisite-free defaults (cf [Bra93]).

3 Reasoning with and about PEC rules

Let us define a concept of a derivation for the very general language of PEC rules. Interestingly, it will not only allow us to handle both monotonic and defeasible rules in the same setting, but it will also allow us to derive both of them.

A word of warning: In the following, $\vdash$ does not represent an inference relation. $\vdash \alpha$ (resp. $\nvdash \alpha$) means that α has (resp. does not) the status "inferred" *within the derivation*. Also, "not inferred within the derivation" does not mean "whose negated form cannot be inferred using the inferred formulas occurring in the derivation" (which is a weaker and less interesting notion). *A word of terminology:* $\vdash \alpha$ and $\nvdash \alpha$ are said to be *signed formulas*. Most naturally, $\vdash \gamma$ (resp. $\nvdash \gamma$) is said to be positive (resp. negative).

Definition 3. *Let Γ be a set of unary PEC rules and $\aleph$ be a PEC rule $(\rho, \{\epsilon_1, \ldots, \epsilon_n\}, \varsigma)$. A derivation of $\aleph$ from Γ is a tree T whose nodes are signed formulas such that*

1. *for each leaf of the form $\vdash \alpha$, either $(\alpha_1, \emptyset, \alpha_2) \in \Gamma$ and $\alpha = \alpha_1 \supset \alpha_2$, or $\alpha = \rho$,*
2. *for each leaf of the form $\nvdash \beta$,*
   $\beta \notin Cn(\{\gamma_1 \supset \gamma_2 \mid (\gamma_1, \emptyset, \gamma_2) \in \Gamma\} \cup \{\alpha \mid \vdash \alpha$ is a node of $T\})$,
3. *if $\nvdash \beta$ is a node then it is a leaf,*
4. *each node, if not a leaf, has a tuple $(\vdash \alpha_1, \ldots, \vdash \alpha_k, \nvdash \beta_1, \ldots, \nvdash \beta_m)$ as its parents ($k > 1$ only if $m = 0$),*[1]

[1] This restriction is due to considering only unary rules in our presentation, it is lifted in the general case.

5. *if* $\vdash \alpha$ *is a node whose parents are a tuple* $(\vdash \alpha_1, \ldots, \vdash \alpha_k)$ *then* $\alpha \in Cn(\{\alpha_1, \ldots, \alpha_k\})$,
6. *if* $\vdash \alpha$ *is a node whose parents are a tuple* $(\vdash \alpha_1, \nvdash \beta_1, \ldots, \nvdash \beta_m)$ *then* $(\alpha_1, \{\beta_1, \ldots, \beta_m\}, \alpha) \in \Gamma$,
7. $\rho \in \{\alpha \mid \vdash \alpha$ *is a leaf of* $T\} \cup \{\top\}$, *and* $\{\epsilon_1, \ldots, \epsilon_n\} = \{\beta \mid \nvdash \beta$ *is a node of* $T\}$, *and* $\vdash \varsigma$ *is the root of* T.

We write $\Gamma \mid\!\sim^{\{\epsilon_1, \ldots, \epsilon_n\}} \aleph$ *and, whenever* $\{\epsilon_1, \ldots, \epsilon_n\}$ *is empty,* $\Gamma \mid\!\sim \aleph$.

Let us provide some intuitions and examples. Let us start with the simple case of classical logic: items 2, 3, and 6 are ineffective because there are no negative nodes, while items 4 and 7 gets simpler for the same reason. In fact, the derivation then is a classical proof: it contains only positive nodes and merely displays classical deductive steps. Thus, the conclusion is a PEC rule $\aleph = (\rho, \emptyset, \varsigma)$ where ς is the root of the derivation tree and ρ either is $\top$ or is a formula from a rule of Γ representing a fact (cf $\top \supset a$ in Example 5) or is an extra formula that plays the rôle of an additional hypothesis (cf. a in Example 6, $a \supset b$ in Example 7, $\neg a$ in Example 8). Still in the case that no exception is mentioned, classical trivialization from inconsistency threatens (cf Example 8).

Example 5. Let $\Gamma = \{(a, \emptyset, b), (\top, \emptyset, a), (b, \emptyset, c)\}$.The tree 3.1 is a derivation of $(\top \supset a, \emptyset, c)$ and $(a \supset b, \emptyset, c)$ from Γ. The tree 3.1 is also a derivation of $(\top, \emptyset, c)$ from Γ (please note that the first part of item 7 is satisfied by $\rho \in \{\top\}$).

Example 6. Let $\Gamma = \{(a, \emptyset, b), (b, \emptyset, c)\}$.The tree 3.2 is a derivation of $(a, \emptyset, c)$ from Γ (please note that a plays the rôle of an additional hypothesis). The tree 3.2 is also a derivation of $(a, \emptyset, c)$ from $\Gamma \cup \{(\top, \emptyset, a)\}$. Please compare with the tree 3.1 being a derivation of $(\top \supset a, \emptyset, c)$ from $\Gamma \cup \{(\top, \emptyset, a)\}$.

Example 7. Let $\Gamma = \{(\top, \emptyset, a), (b, \emptyset, c)\}$.The tree 3.1 is a derivation of $(a \supset b, \emptyset, c)$ from Γ.

Example 8. Let $\Gamma = \{(\top, \emptyset, a)\}$. The tree 3.3 is a derivation of $(\neg a, \emptyset, c)$ from Γ.

$$\dfrac{\dfrac{\dfrac{\vdash \top \supset a}{\vdash a} \quad \vdash a \supset b}{\vdash b} \quad \vdash b \supset c}{\vdash c} \qquad (3.1)$$

$$\dfrac{\dfrac{\vdash a \quad \vdash a \supset b}{\vdash b} \quad \vdash b \supset c}{\vdash c} \qquad (3.2)$$

$$\dfrac{\vdash \neg a \quad \dfrac{\vdash \top \supset a}{\vdash a}}{\vdash c} \qquad (3.3)$$

If some PEC rules in Γ have a non-empty set of exceptions, derivation trees may capture reasoning under some proviso(s) (meaning that there are possible exceptions) as can be seen in Example 9. Item 2 guarantees reasoning to be consistent in the sense that exception-free information from Γ (that may, or may not, occur as positive nodes) does not yield exceptions whose absence is required for the reasoning developed to be acceptable (cf Example 10 with Cn being classical logic and Example 11 with Cn being an arbitrary logic). This needs not prevent trivialization (in which case only derivations with no negative node may exist). Item 3 indicates that consistency statements occur as hypotheses, they are not inferred. Item 4 makes sure that each node, if not a leaf, is inferred from exception-free information and/or consistency hypotheses. As to items 5 and 6, they state that only inference steps from Cn and rules (with exceptions) in Γ may apply. Lastly, item 7 specifies what components the PEC rule derived consists of:

– Its conclusion is the root of the derivation tree.
– Its exceptions exhaust all consistency hypotheses occurring in the derivation tree (cf Example 9).
– Its premise, if not $\top$, either amounts to some exception-free statement represented by a rule from Γ, or it is an extra formula that plays the rôle of an additional hypothesis in the reasoning (cf Example 9).

Example 9. Let $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c)\}$.The tree 3.4 is a derivation of $(a \wedge b, \{d, e\}, c)$ from Γ (please note that ρ is $a \wedge b$ that plays the rôle of an additional hypothesis and that $\{d, e\}$ exhausts all negative nodes of the derivation tree). The tree 3.4 is not a derivation of $(a \wedge b, \{d, e, g\}, c)$ from Γ (item 7 in the definition of a derivation fails because g is listed as an exception of the PEC rule derived but $\nvdash g$ is not a node of the derivation tree). The tree 3.4 is not a derivation of $(a, \{d, e\}, c)$ from Γ (here, item 7 is failed for a different reason: the purported ρ is a but $\vdash a$ is not a leaf of the derivation tree).

Example 10. Let $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c), (\top, \emptyset, \neg f)\}$.

If Cn is taken to be classical logic, the tree 3.4 is not a derivation of $(a \wedge b, \{d, e\}, c)$ from Γ. The reason is that item 2 in the definition of a derivation fails as follows. First, $\vdash f$ is a node of the derivation tree hence $f \in \{\alpha \mid \vdash \alpha \text{ is a node of } T\}$. Second, $(\top, \emptyset, \neg f)$ belongs to Γ hence $\top \supset \neg f \in \{\gamma_1 \supset \gamma_2 \mid (\top, \emptyset, \gamma_1 \supset \gamma_2) \in \Gamma\}$. Third, item 2 then becomes $\beta \notin Cn(\{f, \ldots, \top \supset \neg f\})$ that must be checked for β being d and e. However, as Cn is classical logic, $Cn(\{f, \ldots, \top \supset \neg f\})$ contains all formulas of the language, among them are d and e.

Example 11. Let $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, d), (d, \{\neg c\}, c)\}$. The tree 3.5 is not a derivation of $(a \wedge b, \{d, e, \neg c\}, c)$ from Γ because item 2 is failed. First, $\vdash d$ is a node of the derivation tree hence $d \in \{\alpha \mid \vdash \alpha \text{ is a node of } T\}$. As $\nvdash d$ is a leaf, $\beta \notin Cn(\{\alpha \mid \vdash \alpha \text{ is a node of } T\})$ must be checked for β being d and failure is clear.

$$\dfrac{\dfrac{\vdash a \wedge b \quad \nvdash d \quad \nvdash e}{\vdash f} \qquad \vdash f \supset c}{\vdash c} \qquad (3.4)$$

$$\dfrac{\dfrac{\vdash a \wedge b \quad \nvdash d \quad \nvdash e}{\vdash f} \qquad \dfrac{\vdash f \supset d}{\vdash d} \qquad \nvdash \neg c}{\vdash c} \qquad (3.5)$$

4 A versatile approach

It must be clear that the present work is *not* the definition of a new nonmonotonic logic and its proof theory. Instead, it is the definition of a framework expressive enough to capture an approach to the problem of overriding subsumed rules, and general enough to be instantiated by a number of logical formalisms. Importantly, the concept of a derivation is only a tool towards this aim which can be tailored to the proof theory of various logics.

For instance, and importantly, the above concept of a derivation does *not* match inference in default logic. To start with, there is no notion of an extension. Moreover, there is no counterpart to the requirement that a default *must* apply whenever it can. Also, it happens that derivations exist although there is no extension (cf Example 12).

Example 12. Let $\Gamma = (\Delta, \Sigma)$ be a default theory with $\Delta = \{\frac{\top:a}{\neg a}, \frac{b:d,e}{c}\}$ and $\Sigma = \{b\}$. Let us represent Γ by the PEC rules $\Gamma' = \{(\top, \{\neg a\}, \neg a), (b, \{\neg d, \neg e\}, c), (\top, \emptyset, b)\}$. Γ has no extension because Δ contains the default $\frac{\top:a}{\neg a}$, yet there exists a derivation of the PEC rule $(\top, \{\neg d, \neg e\}, c)$ from Γ', as can be seen from the tree 4.1.

Similarly, Example 13 shows that it may happen that a formula is in no extension although there exists a derivation for it.

Example 13. Let $\Gamma = (\Delta, \Sigma)$ be a default theory with $\Delta = \{\frac{\top:a}{b}, \frac{a:\neg b}{c}\}$ and $\Sigma = \{a\}$. Let us represent Γ by the set of PEC rules $\Gamma' = \{(\top, \{\neg a\}, b), (a, \{b\}, c), (\top, \emptyset, a)\}$. Γ has a single extension, i.e., $E = Cn(\{a, b\})$. Although the formula c is not in E, there exists a derivation of the PEC rule $(\top, \{b\}, c)$ from Γ' as shown by the tree 4.2.

$$\dfrac{\dfrac{\top \supset b}{\vdash b} \quad \nvdash \neg d \quad \nvdash \neg e}{\vdash c} \qquad (4.1)$$

$$\dfrac{\dfrac{\vdash \top \supset a}{\vdash a} \quad \nvdash b}{\vdash c} \qquad (4.2)$$

Still, the concept of a derivation is powerful enough to capture credulous reasoning as modeled by default logic. More precisely, if φ is a formula that belongs to an extension of a default theory Γ then there exists a derivation of $\vdash \varphi$ from the set of PEC rules encoding Γ, which amounts to $\Gamma \mid\!\sim^{\{\epsilon_1,\ldots,\epsilon_n\}} (\top, \{\epsilon_1, \ldots, \epsilon_n\}, \varphi)$.

5 X-derivations

We are now to extend the concept of a derivation by taking into account an additional hypothesis, which, in full generality, can be a PEC rule (with or without exceptions). This full-fledged account is called an X-derivation, the details of which are explained and more generally discussed after the formal definition below.

Definition 4. *Let Γ be a set of PEC rules. Let X be a PEC rule. An X-derivation of $\aleph = (\rho, \{\epsilon_1, \ldots, \epsilon_n\}, \varsigma)$ from Γ is a tree T whose nodes are signed formulas such that*

1. *for each leaf of the form $\vdash \alpha$, either $(\alpha_1, \emptyset, \alpha_2) \in \Gamma \cup \{X\}$ and $\alpha = \alpha_1 \supset \alpha_2$, or $\alpha = \rho$,*
2. *for each leaf of the form $\nvdash \beta$, $\beta \notin Cn(\{\gamma_1 \supset \gamma_2 \mid (\gamma_1, \emptyset, \gamma_2) \in \Gamma\} \cup \{\alpha \mid \vdash \alpha \text{ is a node of } T\})$,*
3. *if $\nvdash \beta$ is a node then it is a leaf,*
4. *each node,if not a leaf, has a tuple $(\vdash \alpha_1, \ldots, \vdash \alpha_k, \nvdash \beta_1, \ldots, \nvdash \beta_m)$ as its parents ($k > 1$ only if $m = 0$),*
5. *if $\vdash \alpha$ is a node whose parents are a tuple $(\vdash \alpha_1, \ldots, \vdash \alpha_k)$ then $\alpha \in Cn(\{\alpha_1, \ldots, \alpha_k\})$,*
6. *if $\vdash \alpha$ is a node whose parents are a tuple $(\vdash \alpha_1, \nvdash \beta_1, \ldots, \nvdash \beta_m)$ then $(\alpha_1, \{\beta_1, \ldots, \beta_m\}, \alpha) \in \Gamma \cup \{X\}$,*
7. *$\rho \in \{\alpha \mid \vdash \alpha \text{ is a leaf of } T\} \cup \{\top\}$, and $\{\epsilon_1, \ldots, \epsilon_n\} = \{\beta \mid \nvdash \beta \text{ is a node of } T\}$, and $\vdash \varsigma$ is the root of T.*

We write $\Gamma \mid\!\sim_X^{\{\epsilon_1,\ldots,\epsilon_n\}} \aleph$ and, should $\{\epsilon_1, \ldots, \epsilon_n\}$ be empty, $\Gamma \mid\!\sim_X \aleph$.

The extra hypothesis X mainly comes into play through items 1 and 6. This means that the PEC rule X is actually regarded as supplementing the set of PEC rules Γ. Accordingly, if $X = (\top, \emptyset, \top)$, then an X-derivation of $\aleph$ from Γ happens to be a derivation of $\aleph$ from Γ (cf Example 14). The rôle of each of the three components of the derived rule $\aleph$ is detailed by item 7. Importantly, item 1 expresses that if a positive leaf (tautologies aside) is not some exception-free information encoded as a PEC rule from Γ then it is the premise of $\aleph$. Similarly to Definition 3, conditional reasoning can be conducted using exception-free information as an extra hypothesis, turning it into a positive leaf. However, the conditional piece can now be the X rule itself (more exactly, an equivalent form) when X represents a formula of classical logic for instance (cf Example 16). When X is a PEC rule $(\varrho, \{\xi_1, \ldots, \xi_h\}, \nu)$ that does have exceptions, if its premise ϱ stands as a positive leaf (i.e., $\vdash \varrho$) not issued from a rule in Γ (that is, there exists no $(\kappa, \emptyset, \zeta)$ in Γ such that $\kappa \supset \zeta$ be ϱ), then ϱ turns out to be the premise of $\aleph$ (cf Example 18).

When X is used in the derivation and that the premise ϱ of X is not a leaf, then ϱ comes from a subproof in the tree (cf Example 19).

In all cases, when X is used in the derivation, its premise occurs (as an hypothesis or an intermediate conclusion) higher in the tree. Therefore, not only is X introduced as an extra hypothesis, but when it is mentioned in the tree, if its premise ϱ does not come from a subproof then $\vdash \varrho$ occurs as a leaf (and is regarded as established); hence ϱ enters the set of premises of $\aleph$ (where $\aleph$ is the PEC rule which is the conclusion of the derivation).

More generally, an X-derivation encompasses conditional reasoning in various forms because it involves consistency hypotheses, it can include an extra rule X, and assumes the premise of $\aleph$ (the PEC rule to be inferred).

Example 14. Let us return to Example 9, i.e., $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c)\}$.

The tree below, reproduced from Example 9, is both a derivation and a $(\top, \emptyset, \top)$-derivation of $(a \wedge b, \{d, e\}, c)$ from Γ.

$$\dfrac{\dfrac{\vdash a \wedge b \quad \not\vdash d \quad \not\vdash e}{\vdash f} \qquad \vdash f \supset c}{\vdash c} \tag{3.4}$$

Example 15. Again, $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c)\}$ as in Example 9.

The tree (3.4) above, reproduced from Example 9, is a $(\top, \emptyset, a \wedge b)$-derivation of $(a \wedge b, \{d, e\}, c)$ from Γ, although in a rather vacuous way because the extra hypothesis $X = (\top, \emptyset, a \wedge b)$ is left unused.

The tree (3.4) is not a $(\top, \emptyset, a \wedge b)$-derivation of $(\top, \{d, e\}, c)$ from Γ. The reason is that item 1 in the definition of an X-derivation is not satisfied because $a \wedge b$ is not of the form $\alpha_1 \supset \alpha_2$ while $\rho = \top$.

In contrast, the tree in the next example is a $(\top, \emptyset, a \wedge b)$-derivation of $(\top, \{d, e\}, c)$ from Γ.

Example 16. Let us still consider $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c)\}$. The following tree is a $(\top, \emptyset, a \wedge b)$-derivation of $(\top, \{d, e\}, c)$ from Γ (informally meaning that assuming $a \wedge b$ allows us to conclude c, unless d or e be the case).

$$\dfrac{\dfrac{\dfrac{\vdash \top \supset a \wedge b}{\vdash a \wedge b} \quad \nvdash d \quad \nvdash e}{\vdash f} \quad \vdash f \supset c}{\vdash c} \tag{5.1}$$

Example 17. Once more, $\Gamma = \{(a \wedge b, \{d, e\}, f), (f, \emptyset, c)\}$.

The tree (3.4) reproduced above in Example 14 is a $(a \wedge b, \{d, e\}, f)$-derivation of $(a \wedge b, \{d, e\}, c)$ from Γ although in a rather vacuous way because $X = (a \wedge b, \{d, e\}, f)$ is in Γ.

Indeed, the tree (3.4) is also a $(a \wedge b, \{d, e\}, f)$-derivation of $(a \wedge b, \{d, e\}, c)$ from Γ' where Γ' is taken to be $\Gamma \setminus \{(a \wedge b, \{d, e\}, f)\}$.

Example 18. Let $\Gamma = \{(f, \{e\}, c)\}$. The following tree is a $(a \wedge b, \{d\}, f)$-derivation of $(a \wedge b, \{d, e\}, c)$ from Γ.

$$\dfrac{\dfrac{\vdash a \wedge b \quad \nvdash d}{\vdash f} \quad \nvdash e}{\vdash c} \tag{5.2}$$

Please observe that the premise of X, namely $a \wedge b$, is not issued from Γ hence is also the premise of $\aleph$ (here, X is $(a \wedge b, \{d\}, f)$ and $\aleph$ is $(a \wedge b, \{d, e\}, c)$).

Example 19. Let $\Gamma = \{(a \wedge b, \{d, e\}, f)\}$. The following tree is a $(f, \{g\}, c)$-derivation of $(a \wedge b, \{d, e, g\}, c)$ from Γ.

$$\dfrac{\dfrac{\vdash a \wedge b \quad \nvdash d \quad \nvdash e}{\vdash f} \quad \nvdash g}{\vdash c} \tag{5.3}$$

Importantly, X-derivations are not meant to be optimal proofs: There is no endeavour as to avoid detours or to impose shortcuts.

Lastly, a concept of consistency can be introduced into the PEC framework.

Definition 5. *Γ is consistent iff $\Gamma \nvdash \bot$.*

As usual, a notion of consistency opens up a choice of negations. Whatever such a choice of a negation $\sim$ for PEC rules, it is likely to be such that both $\Gamma \nvdash \sim R$ and $\Gamma \nvdash \sim\sim R$ while $\Gamma \nvdash R \,\&\, \sim R$ (where & stands for some conjunction of PEC rules, again whatever choice is made there) is possible. Purposedly, we have left out any notion of inferential closure and similarly any subgrouping of consequences, e.g. in forms of extensions *à la* default logic.

6 PEC-Implicants

Definition 6. *Let Γ be a set of unary PEC rules and $R = (\rho, \{\epsilon_1, \ldots, \epsilon_m\}, \varsigma)$ be a unary PEC rule. A unary PEC rule R' is a PEC-implicant of R modulo Γ iff there exists an R'-derivation $\mathcal{D}$ of (ρ, E^*, ς) from Γ such that*

1. $\forall e' \in E^*, \exists e \in \{\epsilon_1, \ldots, \epsilon_m\}$ *s.t.* $e \in Cn(\{e'\})$,
2. $\forall e'' \in \{\epsilon_1, \ldots, \epsilon_m\} \setminus E^*$, $e'' \notin Cn\{\alpha \mid \vdash \alpha$ *is a node of* $\mathcal{D}\}$.

Definition 7. *Let Γ be a set of unary PEC rules. Let R and R' be two unary PEC rules. R' is a strict PEC-implicant of R modulo Γ iff R' is a PEC-implicant of R and R is not a PEC-implicant of R'.*

To simplify matters, Cn stands for classical logic in all of the following examples.

Example 20. Let Γ be empty. Let $R = (a \wedge b, \{\neg d, \neg e\}, c)$ and $R' = (a, \{\neg d\}, c)$.

R' is a PEC-implicant of R modulo Γ. Indeed, the tree 6.1 below is an R'-derivation of $(a \wedge b, \{\neg d\}, c)$ from Γ, $\neg d$ is in the set of exceptions of $\mathcal{R}$ (taking care of item 1), and $\neg e$ does not follow from $\{a, a \wedge b, c\}$ (taking care of item 2).

Example 21. Let $\Gamma = \{(\top, \emptyset, a \supset \neg e), (\top, \emptyset, \neg e \supset f)\}$. Let $R = (a \wedge b, \{\neg d, \neg e\}, c)$ and let $R' = (f, \{\neg d\}, c)$.

Considering the following tree 6.2, R' is not a PEC-implicant of R modulo Γ. Although this tree is an R'-derivation of $(a \wedge b, \{\neg d\}, c)$ from Γ, item 2 from the definition of a PEC-implicant is not satisfied: $\vdash \neg e$ is a node of the tree although $\neg e$ is an exception of R. Informally, R' then fails to subsume R because the way R' is applied when attempting to infer this "neighbouring" version of R involves a case that happens to be an exception to R.

$$\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \quad \not\vdash \neg d}{\vdash c} \qquad (6.1)$$

$$\dfrac{\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \quad \vdash a \supset \neg e}{\vdash \neg e} \quad \vdash \neg e \supset f}{\vdash f} \quad \not\vdash \neg d}{\vdash c} \qquad (6.2)$$

$$\dfrac{\dfrac{\vdash a \wedge b}{\vdash a} \quad \not\vdash h}{\vdash g} \quad \not\vdash \neg d}{\vdash c} \qquad (6.3)$$

$$\dfrac{\dfrac{\top \supset (a \supset b)}{\vdash a \supset b} \quad \vdash a}{\vdash b} \qquad (6.4)$$

$$\dfrac{\vdash a \quad \not\vdash \neg d \wedge \neg f}{\vdash c} \qquad (6.5)$$

Example 22. Let $\Gamma = \{(a, \{h\}, g)\}$. Let $R = (a \wedge b, \{\neg d, \neg h\}, c)$ and let $R' = (g, \{\neg d\}, c)$.

Considering the tree 6.3., R' is not a PEC-implicant of R modulo Γ. Although the tree 6.3 is an R'-derivation of $(a \wedge b, \{\neg d, h\}, c)$ from Γ, item 1 in the definition of a PEC-implicant is failed because h is a formula in E^* from which no formula in $\{\neg d, \neg h\}$ (i.e., $\{\epsilon_1, \ldots, \epsilon_m\}$) can be inferred. Informally, R' then fails to subsume R because the way R' is applied when attempting to infer this "neighbouring" version of R introduces a new exception.

Example 23. Let Γ be empty. Let $R = (a, \{\neg b\}, b)$ and let $R' = (\top, \emptyset, a \supset b)$.

R' is a PEC-implicant of R modulo Γ. Witness, the tree 6.4 is an R'-derivation of $(a, \emptyset, b)$ from Γ, item 1 is trivially satisfied as E^* is empty, and, as regards item 2, $\neg b$ cannot be deduced from $\{a, b, \top \supset (a \supset b)\}$ (i.e., the formulas in the positive nodes of the tree). This example shows that the context is taken into when it comes to assessing whether a rule subsumes another one, in the sense of being a PEC-implicant. Indeed, the implicant and the implicate need not have the same premise.

Example 24. Let Γ be empty. Let $R = (a, \{\neg d, \neg e\}, c)$ and let $R' = (a, \{\neg d \wedge \neg f\}, c)$.

R' is a PEC-implicant of R modulo Γ. Firstly, the tree 6.5 is an R'-derivation of $(a, \{\neg d \wedge \neg f\}, c)$ from Γ. As to item 1, $\neg d \wedge \neg f$ (namely, the only member of E^*) entails $\neg d$ (a member of $\{\epsilon_1, \ldots, \epsilon_m\}$). As to item 2, neither $\neg d$ nor $\neg e$ (the members of $\{\epsilon_1, \ldots, \epsilon_m\}$) are entailed by $\{a, c\}$ (the formulas in the positive nodes of the tree).

Fairly weak requirements about Cn are enough to show that being a PEC-implicant defines a pre-order. Of special interest then is the case that two PEC rules are PEC-implicants of each other: They surely are equivalent in a strong sense closely related to Cn-equivalence of exceptions. It is possible to obtain such a result, as follows.

Given two unary PEC rules $R = (\rho, \{\epsilon_1, \ldots, \epsilon_m\}, \varsigma)$ and $R' = (\rho', \{\epsilon'_1, \ldots, \epsilon'_n\}, \varsigma')$, if R is a PEC-implicant of R' modulo Γ, and R' is a PEC-implicant of R modulo Γ where $\Gamma = \emptyset$ then:

1. $\rho \| \vdash \rho'$ and $\rho' \| \vdash \rho$,
2. $\forall \epsilon_i \in \{\epsilon_1, \ldots, \epsilon_m\}, \exists \epsilon_j, \epsilon'_k$ where $\epsilon_j \in \{\epsilon_1, .., \epsilon_m\}$ and $\epsilon'_k \in \{\epsilon'_1, .., \epsilon'_n\}$, s.t. $\epsilon_j \| \vdash \epsilon_i$ et $\epsilon_j \| \vdash \epsilon'_k$ and $\epsilon'_k \| \vdash \epsilon_j$.

In particular, item 2 means that exceptions in R and R' are the same, up to logical equivalence (by subsumption, there can be more exceptions in R or in R', though).

7 Overriding subsuming rules

We are now ready to introduce our approach to override subsuming rules.

To override the subsuming rules of a PEC rule R and make R preempt, it is presumably not sufficient to "withdraw" all strict PEC-implicants of R and insert R (or "revise" by R in case of inconsistency). Indeed, there may remain in the resulting Γ some information of a self-conflicting change, e.g. so that whenever R is derivable, one of its strict PEC-implicants is also derivable. Accordingly, the process will be a little more elaborate.

In the sequel, we assume two operators $\oplus$ and $\setminus$ to be available in the PEC framework with the following features. Intuitively, $\setminus$ is a kind of contraction operator which applies to a set of PEC rules and to a pair of PEC rules: $\Gamma \setminus (R, R')$ is intended to contract Γ of R' in the presence of R. Intuitively, $\oplus$ is some revision operator in the PEC framework that restores consistency while enforcing means to derive a given PEC rule. More formally, the following properties are required upon these two operators:

- $\Gamma \setminus (R, R') \not\sim^{\varepsilon}_{R} R'$
- $\Gamma \setminus (R, R') \sim^{\varepsilon} R'' \quad \Rightarrow \quad \Gamma \sim^{\varepsilon} R''$
- $\Gamma \setminus (R, R') = Cn(\Gamma \setminus (R, R'))$
- $\Gamma \oplus R$ is consistent
- $\Gamma \oplus R \sim^{\varepsilon} R$

whenever R, R', R'' are PEC rules and Γ is a set of PEC rules that does not need to be consistent.

Definition 8. *Let R' be a strict PEC-implicant of R modulo Γ.*
$\Gamma \oplus_{\rangle R' \langle} R =_{def} \Gamma \setminus (R, R') \cup \{R\}$.

Theorem 1. *Let R' be a strict PEC-implicant of R modulo Γ.*

$$\Gamma \oplus_{\rangle R' \langle} R \not\vdash^{\varepsilon} R'.$$

$$\Gamma \oplus_{\rangle R' \langle} R \vdash^{\varepsilon} R.$$

The next step consists in iterating the above process on all strict implicants of X. Assuming that the $\setminus$ operator is extended so that it applies to all the elements of its second argument which is now a set of PEC rules, we only need one more definition.

Let $\mathcal{Y}$ be the finite set of strict PEC-implicants of R modulo Γ.

Definition 9. $\Gamma \oplus_{\rangle \mathcal{Y} \langle} R =_{def} \Gamma \setminus (R, \mathcal{Y}) \cup \{R\}$.

Theorem 2. *$\Gamma \oplus_{\rangle \mathcal{Y} \langle} R \not\vdash R'$ for all R' that is a strict PEC-implicant of R modulo Γ. Also, $\Gamma \oplus_{\rangle \mathcal{Y} \langle} R \vdash R$.*

8 Conclusions and future work

The contribution of this paper is at least twofold. First, a unified framework has been presented that allows both monotonic knowledge and defeasible rules to be represented and reasoned about in a uniform way. Derivation tools have been defined allowing to reason and infer both kinds of knowledge indifferently. The next step will be to address algorithmic aspects of X-derivations and associated inference, within the propositional setting. Also, the X-derivation concept implements the possibility to state defeasible rules as extra assumptions, which are coming in addition to the defeasible character of rules with exceptions. We believe that this two-levels form of hypothetical reasoning could be further explored and refined. Also, a whole family of forms of implicants could be devised for defeasible rules, depending on the actual form of reasoning that is modelled and on the intended actual epistemological rôles of the involved exceptions, premises and conclusion. Second, this framework has been exploited to solve a specific problem in knowledge representation and reasoning that has not received much attention so far. Namely, how could new information override the relevant subsuming available one? We claim that such an issue should not be taken for granted. Indeed, in real life we do often get information that is logically weaker but that appears *more precise* than the previously recorded one, and should therefore be preferred.

References

[AGM85] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.*, 50(2):510–530, 1985.

[Bra93] Stefan Brass. On the semantics of supernormal defaults. In *Thirteenth International Joint Conference on Artificial Intelligence (IJCAI'93)*, pages 578–583, 1993.

[PB91] Tarcisio Pequeno and Arthur Buchsbaum. The logic of epistemic inconsistency. In *Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 453–460, 1991.

[Rei80] Ray Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.